



19 BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENT- UND  
MARKENAMT

12 Patentschrift  
10 DE 198 60 225 C 2

51 Int. Cl.<sup>7</sup>:  
G 06 T 17/00  
G 06 T 9/00  
H 04 L 9/18

21 Aktenzeichen: 198 60 225.1-53  
22 Anmeldetag: 24. 12. 1998  
43 Offenlegungstag: 29. 6. 2000  
45 Veröffentlichungstag  
der Patenterteilung: 8. 3. 2001

DE 198 60 225 C 2

Innerhalb von 3 Monaten nach Veröffentlichung der Erteilung kann Einspruch erhoben werden

<p>73 Patentinhaber: Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V., 80636 München, DE</p> <p>74 Vertreter: Rösler, U., Dipl.-Phys.Univ., Pat.-Anw., 81241 München</p>	<p>72 Erfinder: Benedens, Oliver, 64342 Seeheim-Jugenheim, DE</p> <p>56 Für die Beurteilung der Patentfähigkeit in Betracht gezogene Druckschriften: OHBUCHI, R., MASUDA, H., AONO, M.: Watermaking Three-Dimensional Polygonal Models Through geometric and Topological Modifications, IEEE Journal on Selected Areas in Communications, Vol. 16, No. 4, May 1998, S. 551-560; DISTANTE, A. u.a.: A Model-Based 3-D Vision, System für Bin-Picking, IEEE Trans. On Circuits and Systems, Vol. 35, No. 5, May 1988, S. 545-553; SUN, C., SHERRAH, J.: 3D Symmetry Detection Using The Extended Gaussian Image, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 19, No. 2, February 1997, S. 164-168;</p>
--	--

54 Verfahren zur Markierung eines Polygon-basierten Binärdatensatzes von einem dreidimensionalen Modell

57 Beschrieben wird ein Verfahren zur Markierung eines Polygon-basierten Binärdatensatzes von einem dreidimensionalen Modell durch gezielte Einbettung einer Information in den Binärdatensatz, der das Modell durch eine Vielzahl miteinander vernetzter Flächenelemente, die jeweils durch Flächeneckpunkte definiert sind, beschreibt, durch Veränderung der Lage wenigstens einzelner Flächeneckpunkte.  
Die Erfindung zeichnet sich dadurch aus, daß durch die Lageveränderung einzelner Flächeneckpunkte eine Änderung in der Flächennormalenverteilung hervorgerufen wird, daß eine Vielzahl von Gruppierungen einzelner Flächennormalen zu sogenannten Einbettungs-Primitiven gebildet wird, wobei jedes Einbettungs-Primitiv als Bin bezeichnet wird und durch eine zentrale Normale, das sogenannte Bin-Zentrum und einen Radius definiert ist, wobei jeweils jenen Bins wenigstens eine Flächennormale eines Flächenelementes des Modells zugeordnet wird, deren Winkeldifferenz zu allen Bin-Zentren minimal ist oder die innerhalb eines Bereiches um die zentrale Normale liegen, der durch einen bestimmten Radiuswert R um die zentrale Normale definiert ist, und daß zu jenen Bins, die wenigstens einer Flächennormalen des Modells zugeordnet sind, ein Mittelwert aller Flächennormalen gebildet wird, der zur Einbettung der Information verschoben wird.

DE 198 60 225 C 2

5 Die Erfindung bezieht sich auf ein Verfahren zur Markierung eines Polygon-basierten Binärdatensatzes von einem dreidimensionalen Modells durch gezielte Einbettung einer Information in den Binärdatensatz, der das Modell durch eine Vielzahl miteinander vernetzter Flächenelemente, die jeweils durch Flächeneckpunkte definiert sind, beschreibt, durch Veränderung der Lage wenigstens einzelner Flächeneckpunkte.

10 Stand der Technik

Im Bereich der Publikation von multimedialen Daten in globalen Netzwerken tritt häufig das Probleme des Copyright-schutzes auf. Solche Daten sind prinzipiell leicht und ohne jegliche Verluste an Qualität zu kopieren. Um den Problemen illegaler (Raub-)Kopien oder Nutzung dieser Daten im Widerspruch mit Lizenzbestimmungen zu begegnen, kann ein so-

15 genanntes Wasserzeichen in die Daten eingebettet werden. Man unterscheidet grundsätzlich zwischen sogenannten geheimen und öffentlichen Wasserzeichen.

Ein geheimes Wasserzeichen kann Informationen enthalten, die eine konkrete Kopie und damit etwa den Lizenznehmer oder Käufer identifizieren oder den Nachweis der Urheberschaft in Streitfällen ermöglichen. Um das geheime Wasserzeichen auslesen zu können, werden die geheimen Einbettungsparameter benötigt, im einfachsten Falle ein vom Menschen erinnerbarer Schlüssel. Die Parameter sind nur der einbettenden Instanz bekannt, die Information daher für den Lizenznehmer oder Käufer nicht extrahier- und damit lesbar.

An ein geheimes Wasserzeichen werden hohe Anforderungen in Bezug auf Robustheit gestellt. Hingegen sind die Anforderungen im Bezug auf Kapazität gering zu nennen. Eingebettet werden, wie bereits erwähnt, Seriennummern oder Hashwerte.

25 Hingegen soll ein öffentliches Wasserzeichen von jedem Nutzer bzw. Empfänger von Copyright-behafteten Material gelesen werden können. Es beinhaltet üblicherweise Copyright- oder Lizenzinformation, bspw. den Copyright-Inhaber, den Erzeuger des Materials oder einen Verweis (URL) auf eine Web-Seite von der mehr Informationen erhalten werden können.

Ein öffentliches Wasserzeichen kann eine Seriennummer tragen, die das Material gegenüber Registrierungsinstanzen identifiziert. Es ist allgemein ein Ersatz für Informationen in Headern oder Sektionen in Dateien, die benutzerdefinierte Informationen tragen können und dem Risiko durch Entfernung in Formatkonvertierungen ausgesetzt sind. Die Parameter zum Auslesen des öffentlichen Wasserzeichens sind öffentlich, ggf. standardisiert.

Ein öffentliches Wasserzeichen stellt hohe Anforderungen an das Einbettungs-Verfahren in Bezug auf Kapazität. Weil Sinn und Zweck eines öffentlichen Wasserzeichens die Bereitstellung von zusätzlicher Copyrightinformation für den Empfänger und nicht der Nachweis der Urheberschaft oder das Verfolgen von Kopien ist, sind die Anforderungen im Bezug auf Robustheit vergleichsweise gering.

Zur Einbettung derartiger Informationen in Form von Wasserzeichen sind sogenannte 1-Bit und n-Bit-Verfahren bekannt.

Die Unterscheidung zwischen diesen beiden Verfahrenstypen basiert auf der Frage, die das Verfahren beim Auslesen des Wasserzeichens beantwortet. Ein 1-Bit Verfahren bettet faktisch genau 1 Bit Information ein. Beim Auslesen werden die Fragen "Ist das Wasserzeichen W in den Daten enthalten?" oder "Welches der Wasserzeichen der Menge WS wurde (mit größter Wahrscheinlichkeit) eingebettet?" beantwortet. Ein n-Bit Verfahren beantwortet die Frage "Welche n Bits wurden eingebettet?".

Der Vorteil eines n-Bit Verfahrens ist darin begründet, daß bspw. Seriennummern oder Verweise auf extern gespeicherte Kundenidentitäten an konstanten Positionen eingebettet werden können, so daß der ausgelesene Bit-String einen Kunden direkt identifiziert. Im Falle eines 1-Bit Systems sind Einbettungs-Positionen verschieden für jedes Wasserzeichen, d. h. beim Auslesen müssen zum einen alle potentiellen Wasserzeichen-Kandidaten bekannt sein, zum anderen muß ein Test auf all diese Wasserzeichen erfolgen.

Im Falle des Einbettens einer Seriennummer könnte ein 1-Bit Verfahren folgendermaßen ablaufen:

50 Ein Generator wird mit der Seriennummer initialisiert und liefert Positionen, an denen Modifikationen abhängig von einem bestimmten, globalen Merkmalswert vorgenommen werden, bspw. im Falle von Bitmaps, Modifikation der Varianz der Pixel-Intensitäten an allen Einbettungs-Positionen.

Ein n-Bit Verfahren kann in ein 1-Bit Verfahren transformiert werden, indem verschiedene Positionen für jedes Wasserzeichen gewählt werden und stets ein konstantes Muster als Bit-String eingebettet wird, bspw. "010101 . .".

55 Der Vorteil eines 1-Bit-Systems gegenüber einem n-Bit-System liegt in der höheren Verlusttoleranz. Ein n-Bit System muß Verluste ausschließlich durch Anwendung von fehlerkorrigierenden Codes oder redundantem Einbetten von Information kompensieren.

Die Kapazität von 1-Bit Systemen kann jedoch als unbegrenzt bezeichnet werden, da ein Generator der die Einbettungspositionen liefert mit dem Hashwert einer beliebig langen Information initialisiert werden kann. Allerdings wird diese unbegrenzte Kapazität durch einen "langsamen" Auslesevorgang erkauft (im Mittel muß die Hälfte aller potentiellen (vermuteten) Wasserzeichen getestet werden).

60 In dem Beitrag von R. Ohbuchi, H. Masuda, and N. Aono, "Watermarking Three-Dimensional Polygonal Models", ACM Multimedia 97, 1997, ACM Press, New York, pp. 261-272, der ersten Publikation im Bereich des Einbettens von Wasserzeichen in Polygon-basierte 3D Modelle, werden eine Reihe von Verfahren dargelegt. Diese können grob in Mesh und Topologie-modifizierende Verfahren kategorisiert werden. Bei beiden Verfahrenstypen werden zur Informationseinbettung Lageänderungen an Punktkoordinaten durchgeführt, die in ihrer Gesamtheit das dreidimensionale Modell in Art eines Gitters ergeben. Die Mesh-modifizierenden Methoden, insbesondere das vorgestellte Tetrahedral Volume Ratio (TVR) Verfahren zeigen fast optimale Eigenschaften im Bezug auf die Kriterien Kapazität, Ausführungsgeschwindigkeit

und Monitoring-Eignung. Die entscheidenden Nachteile sind geringe Robustheit gegen Re-Meshing, Operationen wie Polygon-Reduktion und zufälliges Modifizieren von Punkten. Für das Anwendungsgebiet von öffentlichen Wasserzeichen stellen diese Verfahren adäquate Lösungen dar.

Die Topologie-modifizierenden Verfahren, die, vereinfacht gesprochen, Informationen durch das Schneiden von Löchern in Polygonnetze einbetten, sind trivialerweise anfällig gegen Attacken, die nach diesen Löchern suchen.

Das in der vorstehend genannten Publikation beschriebene Mesh Density Pattern (MDP) Verfahren, ein Verfahren das sichtbare Wasserzeichen einbettet, ist jedoch anfällig gegen Remeshing-Attacken, bei denen Flächen mit möglichst identischen Attributen im Bezug auf Winkel und Größe generiert werden.

#### Darstellung der Erfindung

Die Aufgabe der Erfindung ist es ein Verfahren zur Markierung eines Polygon-basierten Binärdatensatzes von einem dreidimensionalen Modells durch gezielte Einbettung einer Information in den Binärdatensatz, der das Modell durch eine Vielzahl miteinander vernetzter Flächenelemente, die jeweils durch Flächeneckpunkte definiert sind, beschreibt, derart auszubilden, daß die eingebettete Information in Art eines Wasserzeichens möglichst robust gegenüber Manipulationen am Binärdatensatz ist.

Das Verfahren soll insbesondere darauf abzielen, möglichst unabhängig von der konkreten Repräsentation des Modells in Form von Punkt-/Flächenkonfigurationen zu arbeiten. Auf diese Weise soll eine Robustheit des Verfahrens, d. h. Stabilität und Auslesbarkeit des Wasserzeichens gegenüber die Punkt-/Flächenkonfiguration verändernden Operationen, als bedeutendste ist Polygon-Reduktion zu nennen, erhöht werden.

Das Verfahren zur Einbettung von Wasserzeichen soll die Realisierung der folgenden zwei generellen Anwendungen erlauben:

- Verfolgen bzw. Wiedererkennen von Kopien eines 3D Modelles durch Einbetten von bspw. einem Lizenznehmer oder Käufer identifizierenden Information, bspw. einer einzigen Seriennummer in jeder Kopie.
- Nachweis der Urheberschaft durch Einbettung des Hash-Wertes des (Wasserzeichen-freien) Originals in jede der Kopien.

Die Lösung der der Erfindung zugrundeliegenden Aufgabe ist im Anspruch 1 angegeben.

Den Erfindungsgedanken vorteilhaft weiterbildende Merkmale sind Gegenstand der Unteransprüche.

Erfindungsgemäß ist ein Verfahren nach dem Oberbegriff des Anspruchs 1 derart ausgebildet, daß durch die Lageveränderung einzelner Flächeneckpunkte eine Änderung in der Flächennormalenverteilung hervorgerufen wird,

daß eine Vielzahl von Gruppierungen einzelner Flächennormalen zu sogenannten Einbettungs-Primitive gebildet wird, wobei jedes Einbettungs-Primitive als Bin bezeichnet wird und durch eine zentrale Normale, das sogenannte Bin-Zentrum und einen Radius definiert ist, wobei jeweils jenen Bins wenigstens eine Flächennormale eines Flächenelementes des Modells zugeordnet wird, deren Winkeldifferenz zu allen Bin-Zentren minimal ist oder die innerhalb eines Bereiches um die zentrale Normale liegen, der durch einen bestimmten Radiuswert R um die zentrale Normale definiert ist, und daß zu jenen Bins, die wenigstens einer Flächennormalen des Modells zugeordnet sind, ein Mittelwert aller Flächennormalen gebildet wird, der zur Einbettung der Information verschoben wird.

Die zentrale Idee des Verfahrens ist durch die Erkenntnis motiviert, daß die Operationen

- Zufälliges Bewegen von Punkten
- Modifikation des Polygonnetzes (der Punkt-Flächen-Konfigurationen)
- Polygon-Reduktion

große Änderungen bzgl. der Punkt-/Flächenkonfiguration des Modells bewirken können.

Ein Polygon-basiertes 3D Modell kann als Sammlung von Flächen unterschiedlicher Größe und Orientierung angesehen werden. Für ein gegebenes Modell existiert eine schier unendliche Zahl unterschiedlicher Polygonnetze die das gegebene Modell approximieren, wobei eine annähernd identische (wahrgenommene) Qualität realisiert wird.

Die erfindungsgemäße Idee besteht darin, Sammlungen von Flächen als Einbettungsprimitive zu verwenden. Wenn die konkrete Repräsentation eines Modells sich als Folge der genannten Operationen ändert, so muß die neue Punkt-/Flächenkonfiguration doch zumindest globale Flächen-Charakteristika bzgl. Größe, Orientierung und Krümmung, wenigstens für die perzeptuell bedeutendsten Merkmale bewahren, andernfalls ist die (wahrgenommene) Qualität substantiell herabgesetzt.

Um Informationen in den Binärdatensatz für ein 3D-Modell einzubetten, induziert das Verfahren konkret Änderungen in der Flächennormalenverteilung des Modells. Es benötigt als Eingabe eine Polygonrepräsentation des Modells, die aus Dreiecksflächen besteht. Das Original-Modell sei mit M bezeichnet. Ein Wasserzeichen W, ein Bit-String beliebigen Inhalts und Länge wird in die Polygonrepräsentation durch Bewegen von Eckpunkten, was wiederum gewisse Änderungen in der Normalenverteilung induziert, eingebettet. Die resultierende Polygonrepräsentation M besteht wiederum aus Dreiecks-Flächen und weist keine Unterschiede zum Original bezüglich Topologie und Adjazenzen (Kanten, Flächen) auf.

Bevor der Einbettungs-Prozeß im Detail beschrieben wird, werden die Begriffe Bin und Extended Gaussian Image (EGI) definiert und erklärt.

Wie bereits erwähnt, benutzt das Verfahren Gruppen von Flächenelementen, die zu sogenannten Einbettungs-Primitive zusammengefaßt werden. Die Flächenelemente werden dabei wie folgt zu Gruppen zusammengefaßt:

Ausgehend von wenigstens einer zentralen Normalen, die auf verschiedenste Weise generiert werden kann, wird um diese Normale, die auch als Bin-Zentrum bezeichnet wird, ein fiktiver Kegel mit einem maximalen Radius R konstruiert. Fallen nun Flächennormalen der Flächenelemente innerhalb des Kegels, so werden diese Flächennormalen und damit

verbunden diese Flächenelemente dieser Normalen, dem sogenannten Bin zugeordnet, was auch zugleich der Gruppierung der Flächenelementen entspricht. Wichtig ist, daß eine Flächennormale nicht zwei Bins gleichzeitig zugeordnet werden kann, weshalb man bei den Gruppierungen der Flächennormalen von der Bildung disjunkter Mengen spricht. Ein Bin stellt somit das Einbettungs-Primitiv für ein Bit Information dar.

- 5 Die Entscheidung, welchem Bin eine Flächennormale des Modells zugeordnet wird, bestimmt sich durch deren Winkeldifferenz mit allen Bin-Zentren. Die Flächennormale wird dem Bin mit der geringsten Normalen-Differenz zugeordnet.

Die Bins können auf folgende Art und Weise konstruiert werden:

- 10 Die Einheits-Sphäre wird durch Projektion eines weiter unterteilten regulären Polyhedrons, Platonischer Körper, tessaliert. Dies resultiert in Bins mit identischen Flächengrößen und Winkeln. Diese Eigenschaften sind nützlich im Bezug auf die Bestimmung der Orientierung eines Objektes, stellen aber keine Notwendigkeit im Bezug auf den Einbettungsvorgang dar.

Alternativ kann ein Zufallszahlengenerator eine Sequenz von Bin-Zentren (Normalen) mit zugehörigen gleichen oder unterschiedlichen Radien liefern. Es ist sicherzustellen, daß sich die Bins (radialer Einzugsbereich) nicht überlappen.

- 15 Eine graphische Entsprechung des geschilderten Vorgangs der Gruppierung von Flächennormalen des Modells in Bins stellt die diskrete Approximation des Extended Gaussian Image (EGI) dar, was als Orientation Histogramm bezeichnet wird. Siehe auch: B. K. P. Horn, "Robot Vision", The MIT Electrical Engineering and Computing Series, MIT Press, Cambridge, Mass., 1986 oder K. Ikeuchi, "Determining Attitude of Object From Needle Map Using Extended Gaussian Image", Massachusetts Institute of Technology, A. I. Memo No. 714, Cambridge, Mass., Apr. 1983.

- 20 Jeder Vektor korrespondiert mit einem Bin-Zentrum: Seine Richtung korrespondiert mit der Bin-Zentrums-Normalen, seine Länge mit der Summe der Flächen des Modells, deren Normalen dem Bin zugeordnet wurden. Siehe hierzu **Fig. 1a** und **1b**. Das **Fig. 1a** zeigt das Orientation Histogramm der Wagner-Büste. Die Bins wurden durch Projektion eines Icosahedrons auf die Einheitskugel und 2-fache weitere Unterteilung in Dreiecke konstruiert. Dies lieferte 320 Bin-Zentren. Der große nach unten gerichtete Normalenvektor wird durch große Dreiecksflächen hervorgerufen, die den Boden der

- 25 Büste bilden. Die Originalbüste von Wagner ist in **Fig. 1b** dargestellt.

Der Einbettungs- und Lesevorgang besteht somit aus mehreren Schritten:

Abfolge der Schritte beim Einbettungsvorgang:

E1: Berechnung "konsistenter" Flächennormalen

E2: Sammlung von Flächennormalen in Bins

- 30 E3: Anwendung des Kern-Einbettungs-Algorithmus

Abfolge der Schritte beim Lesevorgang:

R1: Berechnung "konsistenter" Flächennormalen

R2: Transformierung des Modells in die Spherische Repräsentation (EGI) und Re-Orientierung des Modells

R3: Sammlung von Flächennormalen in Bins

- 35 R4: Anwendung des Kern-Lese-Algorithmus

Zum Einbetten eines geheimen Wasserzeichens

#### Schritt E1

40

Das Verfahren benötigt konsistente Flächennormalen. Modellierer oder Angreifer können Modelle generieren, deren Flächen-Normalen nicht nach außen, sondern, im Extremfall zufällig nach Innen oder Außen weisen. Benötigt werden nicht notwendigerweise korrekt orientierte Normalen, z. B. alle Normalen weisen nach Außen, sondern eine Verfahrensweise, die für ein identisches Modell, stets identische Normalen-Richtungen liefert.

- 45 Eine mögliche Verfahrensweise, die stets konsistente Normalen liefert, ist folgende: Zunächst wird der Schwerpunkt des Modells (center of mass) berechnet. Nun wird von diesem Schwerpunkt aus ein Strahl durch den Mittelpunkt einer jeden Fläche geschickt und dessen Winkel zu den beiden möglichen Flächennormalen ermittelt. Die Normale, die den größeren Winkel aufweist, wird als Flächennormale gewählt. Die Dreieckspunkte werden entsprechend dieser Normalen im Gegen-Uhrzeigersinn angeordnet. Für streng konvexe Objekte zeigen die auf diese Weise berechneten Normalen alle

- 50 nach Außen.

#### Schritt E2

Im Folgenden wird auf normalisierten Vektoren operiert

55

$N_B$  sei die Anzahl Bins,  $BC_i \in \mathbb{R}^3$  die Bin-Zentren und  $R_i \in [0, \dots, \pi/2]$  die Radien der Bins (Winkel gemessen im Bogenmaß) ( $i = 1, \dots, N_B$ ). Jedem Bin  $i$  ( $i = 1, \dots, N_B$ ) werden alle Flächennormalen zugewiesen deren Winkeldifferenz mit der Zentrums-Normalen kleiner als  $R_i$  ist.  $BN_i$  sei die Anzahl Flächennormalen,  $BP_{ij} \in \mathbb{R}^3$  ( $j = 1, \dots, BN_i$ ) seien die Flächennormalen die dem Bin  $i$  zugeordnet werden.

- 60 Weiterhin sei vereinbart, daß sich die Bins nicht überlappen. Dies kann einfach dadurch erreicht werden, daß für eine vorgegebene Menge von Bin Zentren für jeden Bin jeweils ein Radius gewählt wird, der die Hälfte des minimalen Abstands des Bin-Zentrums zu allen anderen Zentren beträgt.

#### Schritt E3

65

Um ein Bit Information einzubetten, wird der Mittelwert aller Normalen eines Bins, auch bezeichnet als center of mass, im Folgenden Massenschwerpunkt genannt, in eine bestimmte Richtung bewegt. Die Einbettung eines Bit-Strings der Längen erfordert eine Verschiebung der Mittelwerte in  $n$  Bins. Diese Modifikation wird konkret durch die Verschie-

bung von Eckpunkten realisiert, was eine Modifikation der Flächennormalen und damit eine Verschiebung des jeweiligen center of mass induziert.

Die Inhalte eines Bins können von einer 3D in eine 2D Repräsentation transformiert werden, die das Bin-Zentrum zum Ursprung hat und den Einheitskreis als Bin-Grenze. Da die 2D Repräsentation zur Verdeutlichung des Einbettungs- und Auslese-Algorithmus vorteilhaft ist, wird die notwendige Transformation im weiteren beschrieben. 5

Es wird nun die Transformation für den Bin mit Index  $i$  ( $i = 1, \dots, N_B$ ) beschrieben. Zunächst werden zwei Vektoren  $X_1, X_2 \in \mathbb{R}^3$  berechnet, die senkrecht zueinander sind und durch das Bin-Zentrum  $BC_i$  verlaufen. Diese Vektoren sind die Hauptachsen des 2D Koordinaten-Systems:

$BC_i$  wird durch zwei aufeinanderfolgende Rotationen um die x- und z-Achse auf die y-Achse rotiert. Die Inverse dieser Transformation wird auf die x- und z-Achsen angewandt. Das Ergebnis sind die Hauptachsen  $X_1$  und  $X_2$  des 2D-Koordinatensystems. 10

Als nächstes werden die Bin-Inhalte, normalisierte 3D Vektoren  $BP_{ij}$  ( $i = 1, \dots, N_B, j = 1, \dots, BN_i$ ), auf folgende Weise in 2D Koordinaten-Paare  $p_{ij} = (x_{ij}, y_{ij})$  transformiert:

$$h = BC_i \cdot BP_{ij}, l_1 = \cos^{-1}(h), P = BP_{ij} - h \cdot BC_i$$
15

Es sei angenommen werden, daß  $h$  zwischen 0 und 1 liegt (die Differenzen zwischen Bin-Zentrum und "zugeordneter" Normalen sei kleiner als 90 Grad).

$$xp = X_1 \cdot BP_{ij}$$

$$yp = X_2 \cdot BP_{ij}$$

$$l_2 = |xp \ yp|$$
20

$$x_{ij} = \begin{cases} xp * \frac{l_1}{l_2} & l_2 > \varepsilon \\ 0 & l_2 < \varepsilon \end{cases}$$
25

$$y_{ij} = \begin{cases} yp * \frac{l_1}{l_2} & l_2 > \varepsilon \\ 0 & l_2 < \varepsilon \end{cases}$$
30

mit geeigneten Werten für  $\varepsilon$  (e. g.  $10^{-6}$ ).

Die Transformation ist näher in **Fig. 2** dargestellt. 35

Bei  $S = s_1, \dots, s_{NB}$ ,  $s_i \in \{0, 1\}$ ,  $i = 1, \dots, N_B$  der in das Modell  $M$  einzubettende Bit-String.

Für jeden Bin  $i$  wird der center of mass,  $com_i = (cx_i, cy_i)$ , berechnet:

$$com_i = \frac{1}{BN_i} \sum_{j=1}^{BN_i} p_{ij}, \quad (i = 1, \dots, N_B)$$
40

Die Distanz zwischen dem Punkt  $p_{ij} = (x_{ij}, y_{ij})$  und dem Zentrum sei als  $r_{init_{ij}}$  bezeichnet:

$$r_{init_{ij}} = \sqrt{x_{ij}^2 + y_{ij}^2}$$
45

Als nächstes wird der ein Optimierungsprozeß beschrieben, der über die Eckpunkte des Modells iteriert:

Der Einbettungsvorgang, ein Optimierungsprozeß, sucht das Massenzentrum eines jeden Bins der 2D Repräsentation, abhängig vom einzubettenden Bit, jeweils (in x-Richtung) nach links (Bit 0) oder rechts (Bit 1) durch iteriertes Bewegen der Eckpunkte zu verschieben. 50

Den Kern des Optimierungsprozesses stellt ein Multidimensional Downhill Simplex Algorithmus dar, der in einem Beitrag von W. H. Press et al., "Numerical Recipes in C – The Art of Scientific Programming", Second Edition, Section 10.4: Downhill Simplex Method in Multidimensions, Cambridge University Press, Cambridge, UK, 1992, pp. 408–412, beschrieben ist. Dieser wird angewandt, um die Nachbarschaft eines jeden Punktes auf ein lokales Kosten-Minimum (bzgl. der einzubettenden Information) zu durchsuchen. 55

In **Fig. 3** ist hierzu eine Darstellung zum Einbetten eines Bits in einen Bin durch Bewegung des Massenschwerpunktes in eine bestimmte Richtung angegeben. Ein 1 Bit ist erfolgreich codiert, wenn der Schwerpunkt in den markierten Bereich des Kreises gelangt.

Der Einbettungsvorgang wird in Form vierer, als Pseudocode formulierter Funktionen, `main()`, `optimizeVertex()`, `costFunc()` und `costs()`, beschrieben. 60

Eine fünfte Funktion, `multidimensionalDownhillSimplex()`, eine Implementierung des multidimensional downhill simplex wird von der Funktion `optimizeVertex()` aufgerufen, um die Umgebung eines Eckpunktes nach einem lokalen Minimum abzusuchen. `multidimensionalDownhillSimplex()` ruft hierbei die Funktion `costFunc()` auf, um die Kosten einer konkreten Verschiebung eines Eckpunktes zu ermitteln.

Die Aufruf-Abhängigkeiten der Funktionen sind folgende: 65

*main ()*

*optimizeVertex ()*

5

*costs ()*

*multidimensionalDownhillSimplex ()*

10

*costFunc ()*

*costs ()*

15 Während des Optimierungs-Prozesses werden nur die Positionen von Eckpunkte modifiziert und zwar in einer Weise, die Topologie, Ecken, Kanten- und Flächenmenge unverändert läßt. Insbesondere werden keinerlei (neue) "Degeneracies" (bspw. Eckpunkt ohne Kante) verursacht.

Es werden Mengen, die die Original-Positionen von Eckpunkten der 3D- und 2D-Repräsentationen enthalten, angelegt.

Zur Funktion main():

20

*$\delta$  = Initialer (radialer) Suchbereich*

*for i=1; i<Anzahl der Iterationen; i++*

25

*for jeden Punkt P des Modells*

*for j=1; j<Anzahl der Verfeinerungen; j++*

*optimizeVertex (P,  $\delta$ )*

30

*reduziere  $\delta$*

$\delta$  wird in multidimensionalDownhillSimplex() ausgewertet. Es beeinflußt den Such-Radius nach dem Kostenminimum bei Bewegung eines Eckpunktes. Nach jeder Iteration durch alle Eckpunkte des Modells wird dieser Wert reduziert. Der angestrebte Effekt ist vergleichbar der Optimierung mit Simulated Annealing Verfahren.

35 Die Funktion main() iteriert mehrere Male durch die Punkte-Menge des Modells. Die Funktion optimizeVertex() wird mit dem aktuellen Eckpunkt P und  $\delta$  als Parametern aufgerufen.

optimizeVertex() versucht nun in der Umgebung von P ein neues lokales Kosten-Minimum zu finden. Die für eine neue Position anfallenden Kosten liefert costFunc(). Wird ein neues Minimum gefunden, wird der Punkt im Mesh (Modell) mit der neuen Position (3D-Koordinaten) aktualisiert.

40 Die Funktion optimizeVertex() iteriert den Optimierungsprozeß für einen Eckpunkt mehrere Male, wobei das jeweils in der Iteration zuvor gefundene neue Minimum den neuen Startpunkt der Suche darstellt.

Zur Funktion optimizeVertex (Vertex P, Factor  $\delta$ ):

initialCosts = costs(P)

P' = multidimensionalDownhillSimplex (initialCosts, P,  $\delta$ )

45

aktualisiere Mesh durch Austausch von P mit P'

Diese Funktion ist lediglich ein Wrapper für die Funktion multidimensionalDownhillSimplex(). Sie berechnet die initialen Kosten, die durch die initiale Position des Punktes P verursacht werden und ruft multidimensionalDownhillSimplex() zur Suche eines neuen (lokalen) Kostenminimums auf. Letztere liefert die neue Minimum-Position oder ruft die alte Position, wenn keine Verbesserung erzielt werden konnte, zurück. Die Eckpunkte werden jeweils mit dem Ergebnis aktualisiert.

50

Die Funktion multidimensionalDownhillSimplex() ruft die Funktion costFunc() auf:

Zur Funktion costFunc (Vertex P):

*maxCosts=0*

55

*if Suchraum verlassen return 2 /\* 2 is max value for costs \*/*

*for alle Flächen-Normalen TN, die mit Punkt P adjazent sind*

60

*if TN nicht in einem markierten Bin enthalten*

*if Differenz TN und originaler Wert von TN >  $\alpha$  return 2*

65

**if** *TN* in einem markierten Bin enthalten

**if** Differenz *TN* und originaler Wert von *TN* >  $\beta$  **or**

*TN* hat (ihren) Bin verlassen **return** 2

**else**

*c* = costs (*TN*)

**if** (*c* > maxCosts) maxCosts = *c*

**return** maxCosts

Sinn und Zweck von costFunc() ist die Prüfung der Verletzung allgemeiner Nebenbedingungen bei der Minimum-Suche:

Wenn allgemeine Bedingungen bezüglich des Such-Raumes und der maximal tolerierten Änderungen von Flächennormalen verletzt werden, wird als Kosten 2 zurückgeliefert, um die Position von einer weiteren Berücksichtigung auszuschließen. Ohne Verletzungen liegen die zurückgelieferten Kosten zwischen 0 und 1.

$\alpha$  und  $\beta$  sind die maximal tolerierten Differenzen für Normalen, die nicht in einem ausgewählten, d. h. zur Einbettung von Bits bestimmten Bin enthalten sind.

costFunc() iteriert über alle (Dreiecks-)Flächen, die mit Punkt *P* adjazent sind und ruft costs() mit den Flächennormalen als Parameter auf.

Zur Funktion costs (Normal *N*):

Schließlich, am Ende der Aufrufkette, berechnet costs(Normal *N*) die Kosten, basierend darauf, wieviel die (aktuelle) Normale *N* zum Ziel beiträgt, den Schwerpunkt in die korrekte Halbebene (oder Halbraum in der 3D Repräsentation) zu bewegen.

Es sei nun angenommen, *N* korrespondiere mit der Normalen  $BP_{ij}$  (die reale Kostenfunktion wird mit mehr Parameter aufgerufen, die den zugehörigen Bin usw. identifizieren).

Bevor die Kosten berechnet werden, wird *N* in das 2D Koordinaten-System abgebildet. Die resultierenden Punkte-Koordinaten seien bezeichnet als  $p'_{ij} = (x'_{ij}, y'_{ij})$ . Die Original-Punktekoordinaten (vor dem Start des Optimierungs-Prozesses) seien mit  $p_{ij} = (x_{ij}, y_{ij})$  bezeichnet.

Die beschriebene Situation ist in **Fig. 3** illustriert.

Die an den Aufrufer zurückgelieferten Kosten *c* werden folgendermaßen berechnet:

$$r = \sqrt{x_{ij}^2 + y_{ij}^2}$$

$$diff = \begin{cases} x_{ij} - x'_{ij} & s_i = 1 \\ x'_{ij} - x_{ij} & s_i = 0 \end{cases}$$

$$c_1 = |r - r_{mit\ ij}|$$

$$c_2 = \begin{cases} 2 & |diff| > \Delta_{\max}, \\ \Delta_{\max} - diff & 0 \leq diff < \Delta_{\max}, \\ 1 & -\Delta_{\max} \leq diff < 0 \end{cases} \quad (1)$$

$$c = w_1 * c_1 + w_2 * c_2$$

mit bestimmten Gewichten  $w_1, w_2 \in [0, 1]$  und  $w_1 + w_2 = 1$ .

Zwei Komponenten beeinflussen die Kosten: Zum einen der Radius des Punktes (Distanz zum Zentrum) in Abhängigkeit zum Originalwert, zum anderen die Verschiebung in *x*-Richtung (Gewinn) die den Massenschwerpunkt beeinflusst, wiederum gemessen am Originalwert.

Ein positiver Betrag der Verschiebung bedeutet, daß der Schwerpunkt in Richtung der richtigen (das gewünschte Bit codierenden) Halbebene verschoben wird. Je größer diese Verschiebung ist, desto geringer die Kosten. Eine Verschiebung in die Gegenrichtung wird mit Kosten von 1 sanktioniert.

$\Delta_{\max}$  ist eine Konstante, die die Größe der Verschiebung in *x* Richtung begrenzt. In Experimenten stellte dies sicher, daß Punkte "ihren" Bin nicht verließen. Ein Überschreiten von  $\Delta_{\max}$  wird mit Kosten von 2 sanktioniert.

Wenn die Differenz der Radien nicht Teil der Kostenfunktion ist, hat dies zur Folge, daß der Algorithmus sich zu "gierig" verhält und Gewinne nahe  $\Delta_{\max}$  zu realisieren versucht, was zur Folge hat, daß die Bin-Inhalte über an den Rand des Bins verschoben werden, von wo sie mit hoher Wahrscheinlichkeit in Folge von Mesh-Änderungs-Operationen den Bin verlassen.

Als optimale Werte haben sich Experimenten, deren Schilderung noch folgt, herausgestellt:

Die optimalen Werte für die Gewichte  $w_1$  und  $w_2$  waren 0.4 und 0.6. Die Anzahl der Iterationen (durch die Punkte-  
menge) war 3, die Anzahl von Verfeinerungen 2,

$\Delta_{\max}$  war 0.3,

5  $\alpha = 5^\circ$ ,  $\beta = 10^\circ$ ,  $\delta$  war konstant durch alle Iterationen hindurch.

Es sei festgestellt, daß  $\Delta_{\max}$  den Betrag einer Normalen-Verschiebung in x-Richtung effektiv auf  $\Delta_{\max} \cdot R_i$  (Radius des  
Bins i) begrenzt.

Nach Beendigung der main() Funktion wurde das Eingabe-Mesh in ein Ausgabe-Mesh transformiert, das den eingebet-  
teten Bit-String S enthält.

10 Nachfolgend wird der Lesevorgang des Wasserzeichens beschrieben:

Um ein eingebettets Watermark extrahieren zu können, muß der Leser über folgende Informationen verfügen: Die An-  
zahl der Bins  $N_B$ , die Bin Zentren (Normalen)  $BC_i$ , ihr Radius  $R_i$  und die originalen center of mass-Werte  $com_i = (cx_i,$   
 $cy_i)$ .

15 Zusätzliche Information, die an dieser Stelle nicht weiter spezifiziert werden soll, wird zur Reorientierung des Modells  
benötigt.

Diese Informationen sind allerdings identisch für alle mit einem Wasserzeichen versehenen Kopien eines Original-  
Modells und müssen daher lediglich einmal gespeichert werden.

Der Lese-Algorithmus kann nun sehr knapp beschrieben werden, da die meisten Schritte mit denen des Einbettungs-  
Algorithmus identisch sind.

20

Berechnung konsistenter Flächennormalen (R1)

Wie in Schritt E1 des Einbettungsverfahrens beschrieben.

25

Re-Orientierung des Modells und Transformierung des Modells in die Sphärische Repräsentation (EGI) und (R2)

Die Transformation wird ausgeführt wie in Schritt E2 des Einbettungsverfahrens beschrieben.

Der Auslesevorgang erfordert die korrekte Orientierung des Modells entsprechend der des Originals.

30

Sammlung von Flächennormalen in Bins (R3)

Wie in Schritt E2 des Einbettungsverfahrens beschrieben.

Die Normalen des Modells werden in  $N_B$  Bins (a priori Daten) gesammelt, wie im Verfahren zur Einbettung beschrie-  
ben.

35

Anwendung des Kern-Lese-Algorithmus (R4)

Die Inhalte der Bins werden von der 3D in die 2D Repräsentation überführt und der Bin-Schwerpunkt (center of mass)  
berechnet. Die Berechnung des Bin-Schwerpunktes erfolgt in der gleichen Weise wie eingangs erläutert. Die so ermittel-  
ten Schwerpunkte seien mit  $com'_i = (cx'_i, cy'_i)$  ( $i = 1, \dots, N_B$ ) bezeichnet.

40

Der Inhalt des Wasserzeichens  $S' = s'_1, \dots, s'_{N_B}, s'_i \in \{0, 1\}$  ( $i = 1, \dots, N_B$ ) wird folgendermaßen berechnet:

$$45 \quad s'_i = \begin{cases} 1 & cx'_i > cx_i \\ 0 & cx'_i \leq cx_i \end{cases} \quad (2)$$

Es wird also lediglich für jeden Bin getestet, ob der Schwerpunkt sich rechts oder links vom Originalwert befindet.

Kompensierung von Verlusten

50

Wird der Algorithmus zur Einbettung beliebiger Informationen (Bit-Strings) verwandt, können folgende Techniken  
zur Kompensation von Verlusten (Bits) eingesetzt werden:

- Erhöhung der Redundanz: n-faches Einbetten des Bit-Strings und jeweils Mehrheits-Entscheidung für jedes Bit
- 55 ("majority voting") im Auslesevorgang.
- Anwendung fehlerkorrigierender Codes.

Die Forderung nach der Möglichkeit der Einbettung beliebiger Bit-Strings besteht für folgende zwei Anwendungs-  
fälle: Einbetten von Information, die den Nachweis der Eigentümerschaft ermöglicht und Einbetten von Public Water-  
marks (bspw. URL oder IP-Adresse einer Web-Site des Copyright-Inhabers, Vertreibers oder Autors).

60

Wenn das Wasserzeichen lediglich eine Verknüpfung zu externer Information darstellt, bspw. beim Einbetten einer den  
Kunden identifizierenden (Serien-)nummer, können folgende Techniken zur Kompensation von Verlusten Anwendung  
finden.

65

- Statt fortlaufender Seriennummern können zufällige Bit-Strings eingebettet werden. Diese sind über externe Ta-  
bellen auf konkrete Seriennummern oder Kundenidentitäten abzubilden. Im Lesevorgang wird der Bit-String als  
Wasserzeichen ausgewählt, der die geringste Hamming-Distanz zum Leseergebnis aufweist.
- Ein Zufallsgenerator kann mit der "einzubettenden" Information selbst initialisiert werden. Die Ausgabe des Ge-



nerators bestimmt nun die Einbettungspositionen, an denen ein konstantes (oder regelmäßiges) Bitmuster, bspw. "010101 . ." eingebettet wird. Während des Auslesevorganges muß nun auf alle möglichen (eingebetteten) Wasserzeichen getestet werden. Ausgewählt wird das Wasserzeichen, das die minimale Hamming-Distanz zum Bitmuster aufweist. Dies stellt ein 1-Bit-Verfahren dar.

5

#### Möglichkeiten der Verbesserung des Wasserzeichenmarkierungs-Verfahrens

Der Algorithmus, wie er in den vorangegangenen Abschnitten beschrieben wurde, bietet folgende Ansatzpunkte für Verbesserungen:

10

- Jeder Normalen in einem selektierten Bin, d. h. ein Bin der 1 Watermark Bit codiert, kann ein Gewicht entsprechend der zugehörigen Oberfläche relativ zur Gesamtfläche des Bins zugeordnet werden. Dieses Gewicht sollte in die Berechnung der Kostenfunktion und des Bin-Schwerpunktes einfließen. Momentan tragen alle Normalen eines Bins zu gleichen Teilen zur Schwerpunkt-Berechnung bei. Wird nun jeder Normalen ein Gewicht zugeordnet, bedeutet dies, eventuell erheblich weniger Flächennormalen zur Verschiebung des Schwerpunktes modifiziert werden müssen. Der Nachteil ist natürlich die "Empfindlichkeit" des Schwerpunktes gegenüber Normalen mit hohen Gewichten. 15
- Bei der Auswahl von Bins zur Einbettung von Informationen, könnten solche bevorzugt werden, deren Normalen sich (überwiegend) in Regionen des Modells befinden, die perzeptuell bedeutende Merkmale darstellen. Ein guter Startpunkt wäre die Selektierung von Regionen, die scharfe Kanten beinhalten oder sich nahe solcher befinden (wobei die Kanten konstituierenden Flächen eine hinreichende Größe aufweisen müssen um als globales perzeptuell bedeutendes Merkmal eingestuft zu werden. 20
- Einbeziehen der Umgebung in die Auswahl eines Bin-Zentrums und seines Radius: Wenn sich die Mehrheit der Normalen eines Bins sich an dessen Begrenzungen und eine große Anzahl von Normalen außerhalb in der Nähe der Begrenzungen, unterliegt der Schwerpunkt des Bins starken Schwankungen. 25
- Reduzierung des Einflusses von Normalen mit einer großen Distanz vom Zentrum auf die Berechnung des Schwerpunktes.
- Wie in Fig. 3 ersichtlich, sind die Regionen zur Codierung 0 und 1 unterschiedlich groß, was zur unterschiedlichen Stabilität von codierten 0 und 1 Bits führt. Dieses Situation kann dadurch begegnet werden, daß die Bin Inhalte so um das Bin-Zentrum rotiert werden, daß der Masse-Schwerpunkt auf der y-Achse zu liegen kommt, was flächengleiche (symmetrische) Regionen liefert. 30

#### Kurze Beschreibung der Erfindung

35

Die Erfindung wird nachstehend ohne Beschränkung des allgemeinen Erfindungsgedankens anhand von Ausführungsbeispielen unter Bezugnahme auf die Zeichnung exemplarisch beschrieben. Es zeigen:

**Fig. 1a** Das Bild zeigt das Orientaion Histogramm der Wagner-Büste. Die Bins wurden durch Projektion eines Icosahe-drons auf die Einheitssphäre und 2-fache weitere Unterteilung in Dreiecke konstruiert. Dies lieferte 320 Bin-Zentren. Der große nach unten gerichtete Normalenvektor wird durch große Dreiecksflächen hervorgerufen, die den Boden der Büste bilden. 40

**Fig. 1b** Wagner Originalbüste,

**Fig. 2** Darstellung zur Transformation der Inhalte eines Bins von 3D nach 2D,

**Fig. 3** Darstellung zum Einbetten eines Bits in einen Bin durch Bewegung des Massenschwerpunktes in eine bestimmt Richtung. Ein 1 Bit ist erfolgreich codiert, wenn der Schwerpunkt in den markierten Bereich des Kreises gelangt. 45

**Fig. 4** Die Original-Büste Wagners mit 11488 Eckpunkten und 22972 Flächen,

**Fig. 5** Büste nach Einbettung eines Wasserzeichens der Länge 16 Bit. Der Bin-Radius war 22 Grad, 16 von 16 gleich-mäßig verteilten Bins wurden ausgewählt. 50

**Fig. 6** Dreiecksflächen, die in ausgewählten Bins enthalten sind in schwarzer Farbe dargestellt, Dreiecksflächen deren Normalen sich um mehr als 0.5 Grad änderten und die nicht in den ausgewählten Bins enthalten sind, sind in grauer Farbe dargestellt. 50

**Fig. 7** Darstellung zeigt die mit einem Wasserzeichen versehene Büste nach Anwendung von pycrunch (die Größe wurde auf 3153 Eckpunkte und 6452 Dreiecksflächen reduziert),

**Fig. 8** Darstellung zeigt die Büste nach Anwendung von plysimplify (2048 Eckpunkte, 4052 Dreiecks-Flächen). In beiden Fällen war das Wasserzeichen (ohne Verlust) lesbar, 55

**Fig. 9** Büste nach Einbettung eines 50 bit langen Wasserzeichens. Die Bin-Größe war 10 Grad, 50 von 100 gleich-mäßig verteilten Bins wurden ausgewählt. Der Bin-Radius war 10 Grad. Die Bedeutung von schwarzen und grauen Fläche ist analog der Fig. 6, und

Tab. 1 tabellarisch aufgelistete Testergebnisse

60

#### Wege zur Ausführung der Erfindung, gewerbliche Verwendbarkeit

Das im vorangegangenen Kapitel dokumentierte Verfahren, wurde in C++ unter Nutzung des Graphik-Paketes MAM/VRS implementiert, siehe hierzu, J. Doellner and K. Hinrichs, MAM/VRS – The Modelling and Animation Machine/ Virtual Renderings System V 2.0, <http://wwwmath.uni-muenster.de/math/inst/info/u/mam>. Die Downhill Simplex Im- 65  
plementierung wurde aus W. H. Press et al., "Numerical Recipes in C – The Art of Scientific Programming", Second Edition, Section 10.4: Downhill Simplex Method in Multidimensions, Cambridge University Press, Cambridge, UK, 1192, pp. 408–412, entnommen und nur leicht modifiziert.

Um die immanente Robustheit des Verfahrens gegen Polygon-Reduktion zu dokumentieren, wurden zwei Polygon-Reduktions-Applikationen, plycrunch und plysimplify auf mit einem geheimen Wasserzeichen versehene Exemplare der Wagner-Büste angewandt. Beide genannten Applikationen sind Bestandteile der Software-Distribution des Simplification Envelopes Verfahrens, siehe hierzu J. Cohen et al., "Simplification Envelopes", Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, ACM Press, New York, 1996, pp. 119–128. sowie Simplification Envelopes V1.2 implementation, <http://www.cs.unc.edu/~geom/envelope.html>.

– plycrunch reduziert die Polygonanzahl eines Objektes durch Zusammenfassen benachbarter Eckpunkte. Diese Methode bewahrt nicht die Topologie des Originals.

– plysimplify reduziert die Anzahl von Polygonen wobei garantiert wird, daß alle Eckpunkte des reduzierten Modells sich in einer vom Anwender spezifizierbaren  $\epsilon$  Umgebung befinden. plysimplify läßt die Topologie des Modells unverändert.

Die beiden Applikation wurden auf drei verschiedene Modelle der Wagner-Büste angewandt. Alle drei Modelle sind Polygon-reduzierte Versionen eines Lasergescannten Originals:  
Modell 1: 3923 Eckpunkte, 7842 Flächen (Dreiecks-)  
Modell 2: 5616 Eckpunkte, 11228 Flächen  
Modell 3: 11488 Eckpunkte, 22972 Flächen

#### Erklärung des Aufbaus von Tabelle 1

Operation ist die angewandte Reduktions-Applikation, si für simplify er für crunch, Eckpunkte und Flächen sind die Anzahl der Eckpunkte und Flächen vor Anwendung der Applikation. Verlust ist der absolute Verlust von Wasserzeichen-Bits (als Folge der Reduktion), Verlust relativ der relative Anteil verfälschter Bits. Reduktion bezeichnet den Quotienten der Anzahl von Flächen vor und nach der Reduktion.

Der Auflistung der Resultate stehen jeweils folgende Informationen voran: Der allen Bins gemeinsame Radius. "m von n" bedeutet, daß die Einheits-Sphäre durch eine Approximation von n gleichmäßig verteilten Bin-Zentren tessaliert wurde, von denen m zufällig zur Einbettung von Watermark-Bits ausgewählt wurden. Die Bins überlappten sich nicht.

Die Bin-Zentren wurden mittels der Applikation Tessel berechnet, siehe Tessel application v2.0, <http://www3.uniovi.es/~quimica.fisica/qcg/tessel/tessel.html>. Diese Applikation tessaliert ein Sphäre mit Punkten, wobei die maximale Distanz von Punkte-Paaren minimiert wird. In den dokumentierten Tests wurden Tessalierungen mit 16,32 und 100 Punkten berechnet. Von diesen wurden zufällig 16 (Tests 1, 3, 4, 5) bzw. 50 (Tests 2, 6) Punkte als Bin-Zentren ausgewählt. In keinem der Testfälle überdeckten die Bins die gesamte Kugeloberfläche.

Tabelle 1 zeigt die Zusammenfassung der Ergebnisse der 6 Testfälle. Fig. 4 bis 6 zeigen die zum Testfall 4 gehörenden Modelle. In diesem Testfall wurde der höchste Grad an Robustheit des Wasserzeichens beobachtet. Allerdings wurde der Grad an Robustheit durch einen Verlust an Betrachtungsqualität erkaufte. Wie in Fig. 5 zu sehen, wurden bspw. die Regionen um den Mund der Büste in Mitleidenschaft gezogen. Einer der Gründe für diesen Verlust resultiert aus der Tatsache, daß wegen der maximalen Bin-Größe (Radius 22 Grad) ein Maximum an Oberflächen des Modells in markierten Bins enthalten und Gegenstand des Einbettungs-Vorganges war.

Ein weiterer Grund resultiert aus der einfachen Tatsache, daß mit zunehmender Anzahl von Elementen in einem Bin, die Anzahl von Kollisionen zunimmt. Kollisionen entstehen, wenn Randbedingungen bezüglich maximal tolerierter Normalenveränderungen verhindern, daß Normalen in Richtung der gewünschten Halbebene bewegt werden können. Verschieberichtungen von Normalen können sich zudem als unvereinbar erweisen.

Für einen kleineren Radius, erhöht sich die Betrachterqualität, allerdings sinkt zugleich die Robustheit, wie anhand des Testfalles 6 in Tabelle 1 und Fig. 9 ersichtlich ist.

Es sei erwähnt, daß sich in allen durchgeführten Tests die Bits nach Einbettung als stabil erwiesen. Dieses Ergebniss war nicht notwendigerweise erwartet worden, da, wie bereits erwähnt, der Einbettungs-Vorgang zahlreiche Kollisionen beinhaltet. Um diese Stabilität zu garantieren, erwies es sich als ausreichend, Bins zu selektieren, die eine gewisse Mindestanzahl von Normalen beinhalteten, um den Normalenaustausch mit der Umgebung (ein Effekt des Einbettungsvorgangs) zu kompensieren.

Das Einbetten eines "0" oder "1" Bits in einen Bin stellt zwei unterschiedliche Optimierungsprobleme dar. Allgemein gesprochen, bedeutet das Einbetten unterschiedlicher Bit-Strings die Lösung unterschiedlicher Optimierungsprobleme. Die Tests zeigten allerdings keinen Anhaltspunkt dafür, daß codierte 0 Bits stabiler als 1 Bits waren und umgekehrt. Diese Eigenschaft ist wichtig im Bezug auf die Einbettung von Bit-Strings an konstanten Positionen.

Es muß betont werden, daß die Eigenschaften der angewandten Reduktions-Methoden nicht bei der Selektierung von Bins berücksichtigt wurden. Die Testergebnisse sollten vielmehr Aufschluß darüber geben, in welchem Maße das Verfahren generell Robustheit gegenüber Polygonreduktion aufweist. Die "crunch"-Ergebnisse bspw. lassen sich signifikant verbessern, indem Bins selektiert werden, deren zugeordnete Flächen vorwiegend in Regionen mit geringer Krümmung lokalisiert sind. Für das Test-Modell waren (Dreiecks-) waren Flächen dieser Regionen vergleichsweise groß, was sich in großen Eckpunkt-Distanzen niederschlug. Letztere Eigenschaft bewirkte, daß ein hoher Prozentsatz von Eckpunkten dieser Bins vom crunch-Algorithmus nicht modifiziert wurde.

#### Patentansprüche

1. Verfahren zur Markierung eines Polygon-basierten Binärdatensatzes von einem dreidimensionalen Modell durch gezielte Einbettung einer Information in den Binärdatensatz, der das Modell durch eine Vielzahl miteinander vernetzter Flächenelemente, die jeweils durch Flächeneckpunkte definiert sind, beschreibt, durch Veränderung der Lage wenigstens einzelner Flächeneckpunkte **dadurch gekennzeichnet**,

daß durch die Lageveränderung einzelner Flächeneckpunkte eine Änderung in der Flächennormalenverteilung hervorgerufen wird,

daß eine Vielzahl von Gruppierungen einzelner Flächennormalen zu sogenannten Einbettungs-Primitive gebildet wird, wobei jedes Einbettungs-Primitive als Bin bezeichnet wird und durch eine zentrale Normale, das sogenannte Bin-Zentrum und einen Radius definiert ist, wobei jeweils jenen Bins wenigstens eine Flächennormale eines Flächenelementes des Modells zugeordnet wird, deren Winkeldifferenz zu allen Bin-Zentren minimal ist oder die innerhalb eines Bereiches um die zentrale Normale liegen, der durch einen bestimmten Radiuswert R um die zentrale Normale definiert ist, und

daß zu jenen Bins, die wenigstens einer Flächennormalen des Modells zugeordnet sind, ein Mittelwert aller Flächennormalen gebildet wird, der zur Einbettung der Information verschoben wird.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß zu jedem Flächenelement eine eindeutige Flächennormale, die sogenannte konsistente Flächennormale, bestimmt wird, indem zur Bestimmung der konsistenten Flächennormale eines jeden Flächenelementes der Schwerpunkt des Modells ermittelt wird, von dem aus jeweils ein Verbindungsstrahl zum Mittelpunkt eines jeden Flächenelementes gebildet wird und jene Flächennormale zu dem jeweiligen Flächenelement als konsistente Flächennormale ausgewählt wird, die einen größeren Winkel zu dem Verbindungsstrahl einschließt.

3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß die Information als Bit-String ausgebildet ist.

4. Verfahren nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, daß zur Einbettung einer Ein-Bit-Information der Mittelwert aller Flächennormalen eines Bins in eine bestimmte Richtung bewegt wird, und daß zur Einbettung eines Bit-Strings der Länge n alle Mittelwerte aller Flächennormalen von mindestens n Bins verschoben werden, wodurch in jedem Fall der Massenschwerpunkt des Modells verschoben wird.

5. Verfahren zur Detektion der mittels des Verfahrens nach einem der Ansprüche 1 bis 4 in einem markierten Binärdatensatzes von einem dreidimensionalen Modell eingebetteten Information, dadurch gekennzeichnet,

- daß der Massenschwerpunkt des markierten Modells ermittelt wird,
- daß zu dem, bereits mit einer Information markierten Binärdatensatzes des eines dreidimensionalen Modells konsistente Flächennormalen bestimmt werden,
- daß der markierte Binärdatensatz derart reorientiert wird, daß das durch den Binärdatensatz gebildete, markierte Modell möglichst deckungsgleich ausgerichtet wird,
- daß eine Gruppierung von Flächennormalen in Bins durchgeführt wird, und
- daß die Schwerpunkte der Bins des markierten Modells bestimmt und mit den Schwerpunkten der Bins des Originalmodells verglichen werden.

6. Verfahren nach Anspruch 5, dadurch gekennzeichnet, daß zur Massenschwerpunktsuntersuchung alle Mittelwerte aller Bins herangezogen werden.

7. Verfahren nach Anspruch 6, dadurch gekennzeichnet, daß zur Ermittlung der markierten Information jeder Bin dahingehend überprüft wird, ob sich sein Mittelwert bzw. Schwerpunkt in der einen oder in der anderen Halbebene bzw. Halbraum des ursprünglichen Schwerpunktes befindet, wobei je nach Halbraum eine Codierung mit 1 oder 0 erfolgt.

8. Verfahren nach einem der Ansprüche 5 bis 7, dadurch gekennzeichnet, daß zur Detektion der markierten Information folgende Werte vorhanden sein müssen:

$N_B$ : Anzahl der zur Beschreibung des Modells erforderlichen Bins

$BC_i$ : Lage der Bin Zentren

$R_i$ : Radien der Bin Zentren sowie

$com_i$ : Mittelwerte der Bins

9. Verfahren nach einem der Ansprüche 5 bis 8, dadurch gekennzeichnet, daß der Vergleich der Schwerpunkte der Bins mit einer 1 zu 1 Zuordnung erfolgt.

---

Hierzu 5 Seite(n) Zeichnungen

---

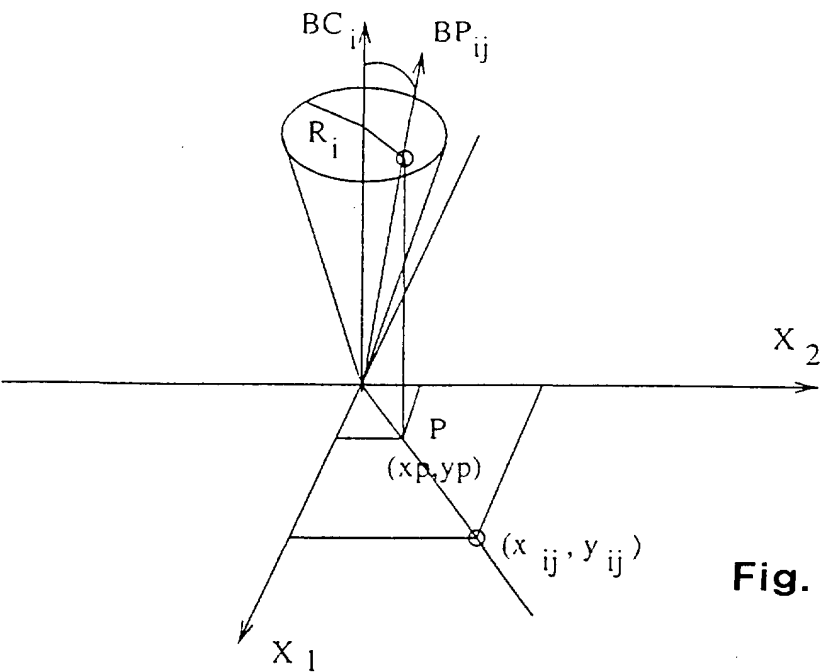


Fig. 2

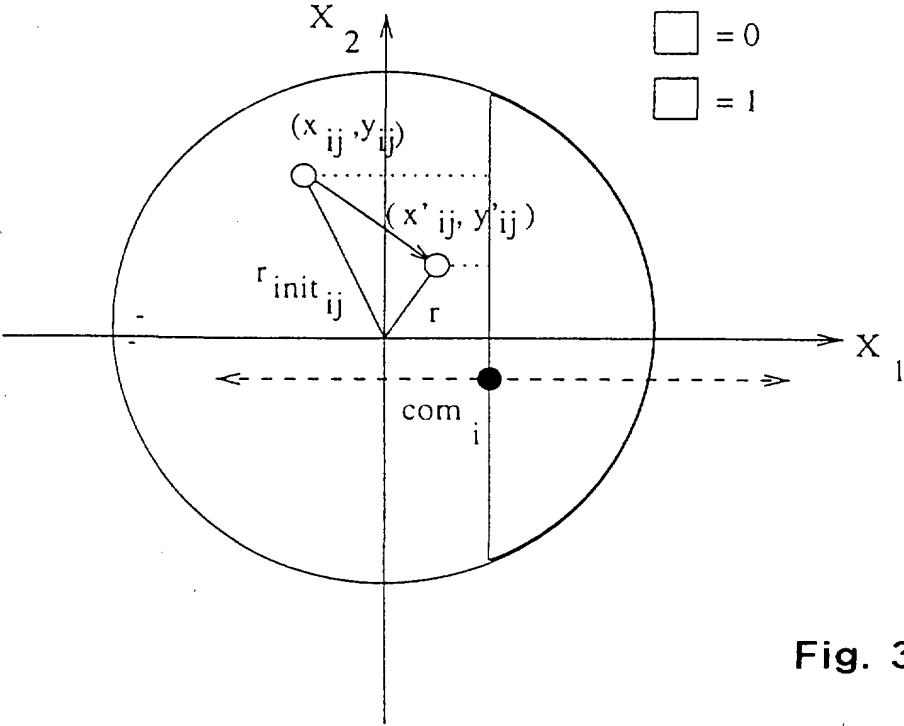
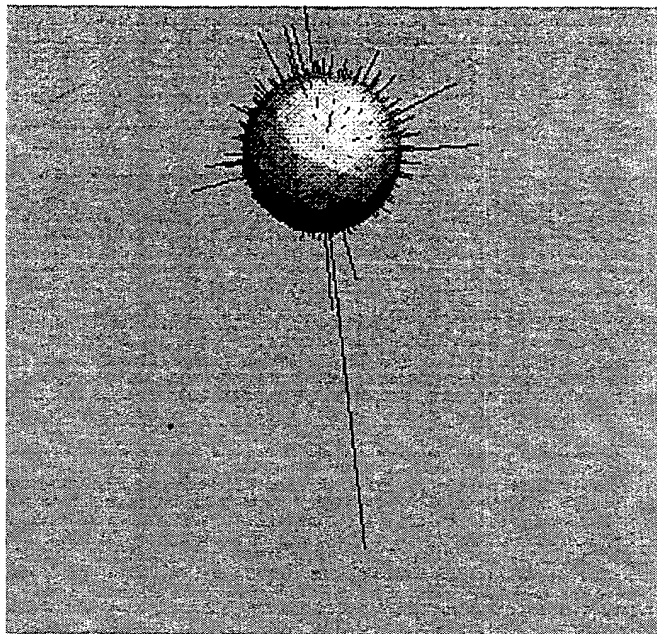


Fig. 3



**Fig. 1a**



**Fig. 1b**



**Fig. 4**



**Fig. 5**



**Fig. 6**



**Fig. 7**



**Fig. 8**



**Fig. 9**

Operation	Eckpunkte	Flächen	Verlust	Reduktion	Verlust relativ
<b>Test no. 1: Modell 1, Bin Radius 20 Grad, 16 von 32 Bins</b>					
si	2643	5282	0	0.67	0.00
si	1769	3534	0	0.45	0.00
si	1580	3156	0	0.40	0.00
si	1243	2482	3	0.32	0.19
si	700	1396	4	0.18	0.25
cr	3326	6654	0	0.85	0.00
cr	2730	5480	2	0.69	0.13
<b>Test no. 2: Modell 1, Bin Radius 10 Grad, 50 von 100 Bins</b>					
si	2913	5822	0	0.74	0.00
si	2641	5278	1	0.67	0.02
si	1451	2898	4	0.37	0.08
<b>Test no. 3: Modell 2, Bin Radius 20 Grad, 16 von 32 Bins</b>					
si	4718	9738	0	0.87	0.00
si	2795	5586	0	0.48	0.00
si	2289	4574	1	0.41	0.06
cr	4661	9380	0	0.84	0.00
cr	3598	7328	1	0.65	0.06
cr	2799	5760	4	0.51	0.25
<b>Test no. 4: Modell 2, Bin Radius 22 Grad, 16 von 16 Bins</b>					
si	2048	4052	0	0.36	0.00
cr	3153	6452	0	0.57	0.00
cr	2473	5076	1	0.45	0.06
cr	2166	4498	3	0.40	0.19
<b>Test no. 5: Modell 3, Bin radius 20 Grad, 16 von 32 Bins</b>					
si	7395	14786	0	0.64	0.00
si	3029	6054	3	0.26	0.19
<b>Test no. 6: Modell 3, Bin radius 10 Grad, 50 von 100 Bins</b>					
si	8353	16666	0	0.72	0.00
si	7467	14930	3	0.65	0.06
si	5991	11978	7	0.52	0.14
si	1743	3482	12	0.15	0.24
si	1617	3220	16	0.14	0.32
cr	9770	19610	2	0.85	0.04
cr	8072	16306	6	0.71	0.12

Tabelle 1